Exercice pratique - PHP5 CRUD

Création d'une application Web de gestion de cinémas (style Allociné)

I. Cahier des charges

A. Contexte

Vous travaillez en tant que développeur indépendant. Le client fait appel à vous pour concevoir et réaliser une application Web permettant de gérer les films à l'affiche de différents cinémas.

B. Architecture technique

L'architecture technique retenue est une architecture 3-tiers. Les technologies utilisées sont les suivantes :

- Couche Présentation
  - HTML5
  - CSS3
  - Javascript 1.8
- Couche Traitements
  - PHP5.6
- Couche Données
  - MySQL 5.6

Le serveur Web à utiliser est Apache2.4.

C. Architecture applicative

Le client demande deux versions de l'application :

- La première version est un POC (Proof Of Concept) à livrer au client pour le 7 Septembre 2015 à 9h00. Elle n'est là que pour montrer au client que vous savez faire.
  - L'architecture MVC n'est pas un pré-requis pour cette version
  - L'approche choisie est l'approche orientée objet
- La deuxième version (version 1.0) disposera des mêmes fonctionnalités mais implémentera en plus l'architecture MVC à livrer au client le 11 Septembre 2015 à 12h00

D. Besoins

L'application peut être d'un point de vue logique séparée en deux parties :

- Gestion des cinémas
- Espace personnel

1. Gestion des cinémas

L'application doit permettre de réaliser toutes les opérations de création, lecture, mise à jour, suppression (CRUD ou Create Read Update Delete) sur les objets métiers suivants :

- Cinéma
- Séance
- Film

Aucune fonctionnalité supplémentaire n'est demandée.

Un cinéma est caractérisé par un nom et une adresse.

Un film est décrit à l'aide de son titre et de son titre original.

Une séance concerne un film donné qui est projeté dans un cinéma donné. Elle possède une heure de début, une heure de fin, une date de projection, la version du film projeté (VF, VO, VOSTFR, VOSTVO).

2. Espace personnel

L'application doit permettre à un utilisateur de se créer un espace personnel dans lequel il pourra sélectionner ses films préférés. L'application doit permettre de réaliser toutes les opérations CRUD sur les objets métiers suivants :

- Utilisateur
- Préfère

Un utilisateur est caractérisé par une adresse courriel, un nom, un prénom et un mot de passe.

Une préférence d'un utilisateur concerne un film en particulier à propos duquel l'utilisateur peut ajouter un commentaire.

3. Droits d'utilisation

Les utilisateurs qui pourront d'inscrire sur le site disposeront des droits suivants **sur les objets métiers qu'ils possèdent** :

- Utilisateur : CRUD
- Préfère : CRUD

Tous les utilisateurs (inscrits comme non-inscrits) disposeront des droits suivants :

- Cinéma : Read
- Film : Read
- Séance : Read

Un super-utilisateur (nommé 'admin') aura tous les droits (CRUD) sur tous les objets métiers de l'application.

E. Méthodologie de développement

Le développement sera itératif et incrémental. Autrement dit, à chaque itération (cycle complet de développement), le nombre de fonctionnalités augmente (on "incrémente" les fonctionnalités).

Chaque livrable à l'issu d'une itération est à faire valider par le client avant de commencer l'itération suivante !

F. Périmètre des versions

Ci-dessous la description des fonctionnalités à développer en fonction du périmètre de chacune des versions.

1. POC

La première itération a déjà été effectuée. Vous n'intervenez qu'à partir de celle-ci.

a) 1ère itération

(1) Espace personnel

- Inscription d'un utilisateur
- Authentification d'un utilisateur (login)
- Logout d'un utilisateur
- Création d'une liste de films préférés pour un utilisateur donné

(2) Gestion des cinémas

N/A

b) 2ème itération

(1) Espace personnel

- Modification (mise à jour / suppression) d'une préférence de film pour un utilisateur donné

(2) Gestion des cinémas

- Consulter la liste des cinémas
- Consulter la liste des films
- Consulter la liste des séances pour un film donné
- Consulter la liste des séances pour un cinéma donné

c) 3ème itération

(1) Espace personnel

N/A

(2) Gestion des cinémas

- Ajouter/Modifier/Supprimer un cinéma
- Ajouter/Modifier/Supprimer un film
- Ajouter/Modifier/Supprimer une séance pour un film et un cinéma donnés

2. Version 1.0

a) 1ère et dernière itération

Migration de l'application selon l'architecture MVC.

G. Qualité de code

En termes de qualité de code, le client est intransigeant. Il souhaite la qualité professionnelle :

- Toutes les pages HTML générées devront respecter les normes définies par le W3C[1]
- Le code PHP se doit de respecter les standards PSR-1[2], PSR-2[3], PSR-3[4] et PSR-4[5].

H. Axes d'améliorations

Après un audit interne, le client mentionne que le livrable de la première itération peut être amélioré notamment au sujet :

- Du test unitaire
- De la gestion des sessions
- Des requêtes à la base de données
    - Certaines requêtes sont exécutées alors qu'elles ne sont pas nécessaires
- De la gestion des logs
    - Le formatage des messages n'a pas été personnalisé
- De la gestion des exceptions
    - De nombreuses exceptions ne sont pas levées
    - Aucune exception personnalisée n'a été implémentée

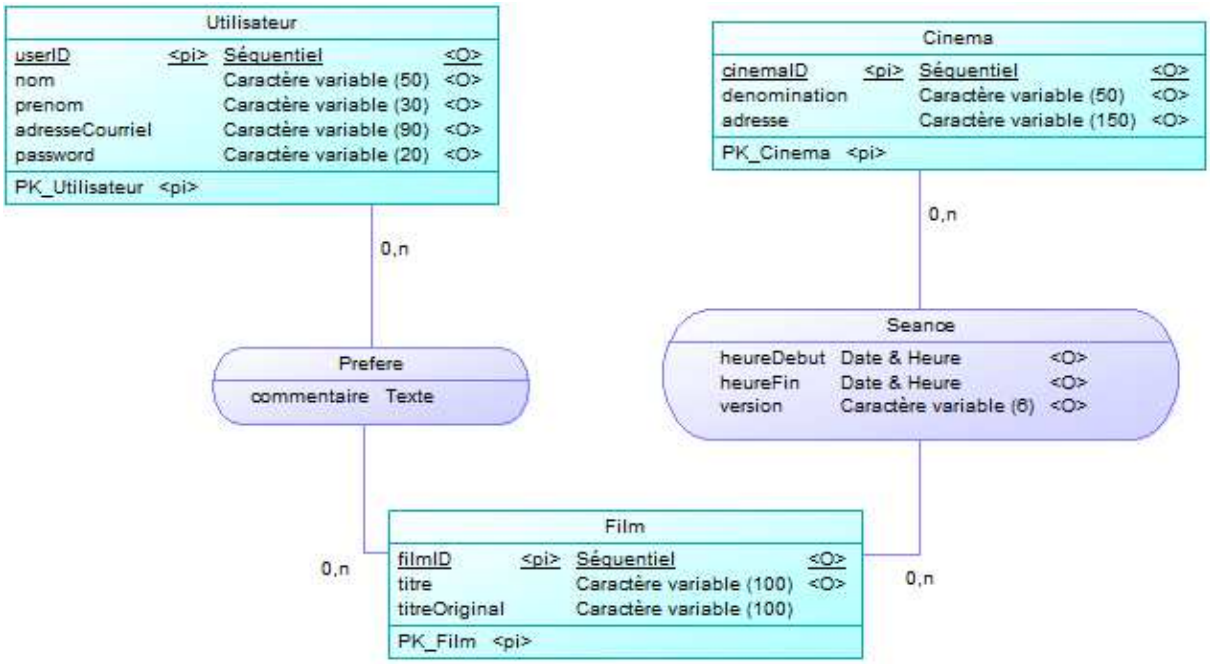
N'hésitez pas à faire part de vos remarques et de vos propositions d'améliorations.

[1] http://validator.w3.org/
[2] https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-1-basic-coding-standard.md
[3] https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-2-coding-style-guide.md
[4] https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-3-logger-interface.md
[5] https://github.com/php-fig/fig-standards/blob/master/accepted/PSR-4-autoloader.md

II. Conception

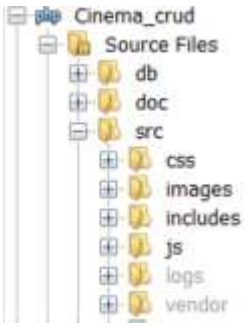
Des éléments de conception vous sont fournis par le client.

A. Modèle Conceptuel de données



B. Arborescence

Le client insiste sur l'arborescence à utiliser :



- db : contient les scripts de création de la BDD
- doc : enoncé de l'exercice
- src : sources de l'application
  - css : contiendra toutes les feuilles de styles de l'application
  - images : contiendra toutes les images utilisées
  - includes : contiendra tous les fichiers PHP inclus dans les pages principales
  - js : contiendra tous les librairies Javascript
  - logs : emplacement des fichiers de consignation
  - vendor : emplacement des librairies tierces utilisées (Composer, Monolog, Psr)

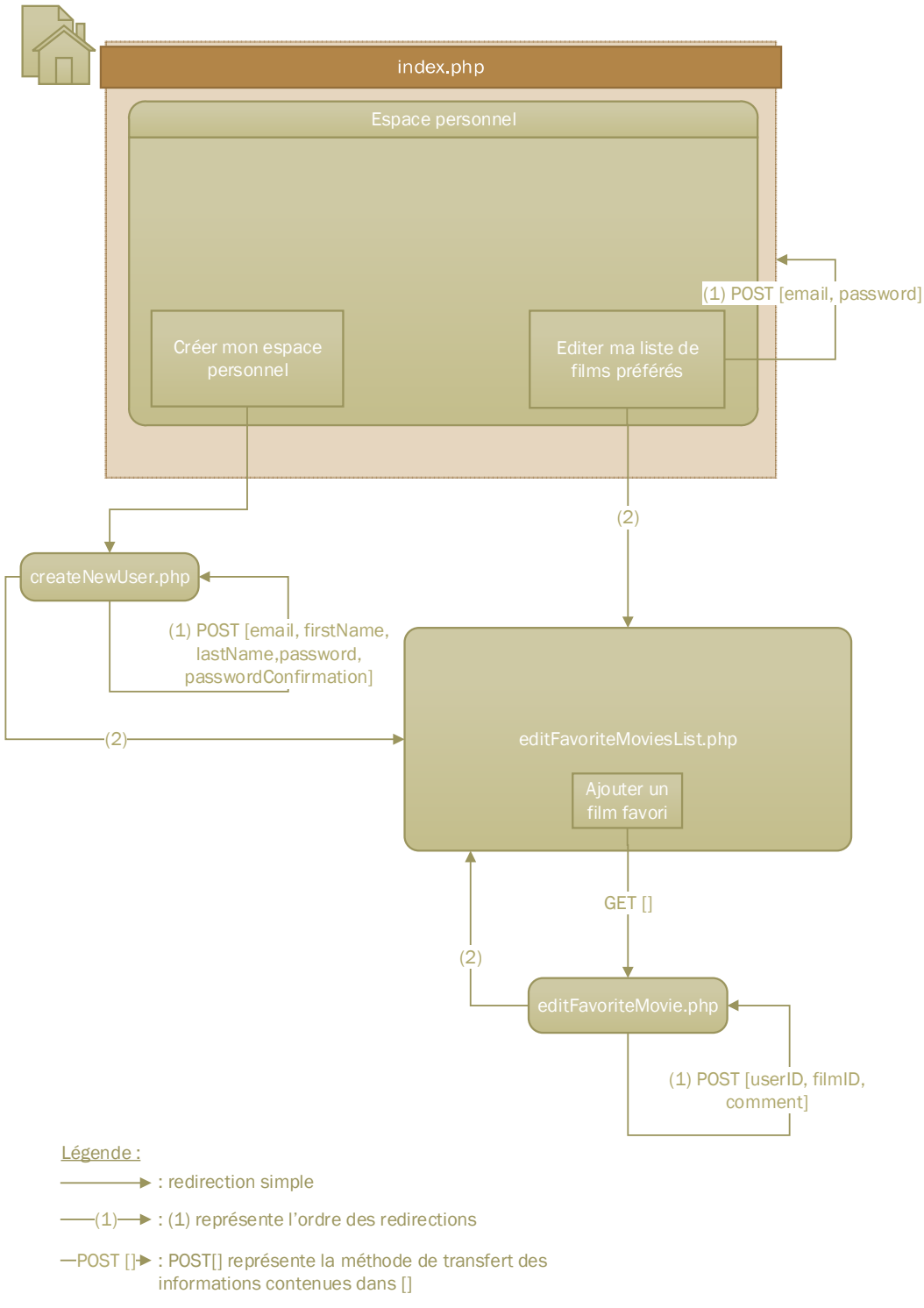
C. Navigation

Ci-dessous, les schémas de navigation de chacune des parties de l'application en fonction des itérations.

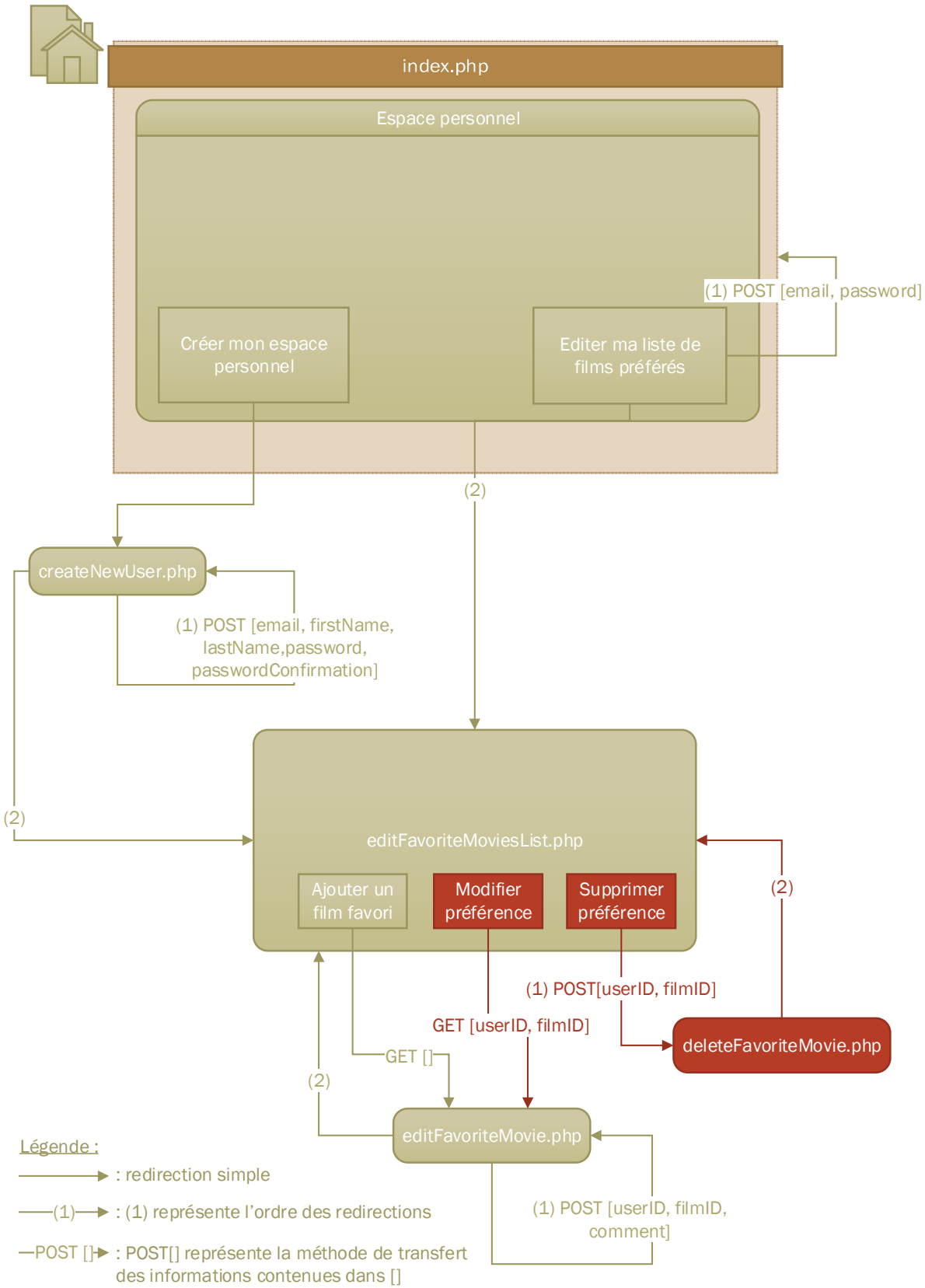
1. Espace personnel

a) POC

(1) 1ère itération



(2) 2ème itération



Les nouvelles fonctionnalités apparaissent en rouge.

(3) 3ème itération

N/A

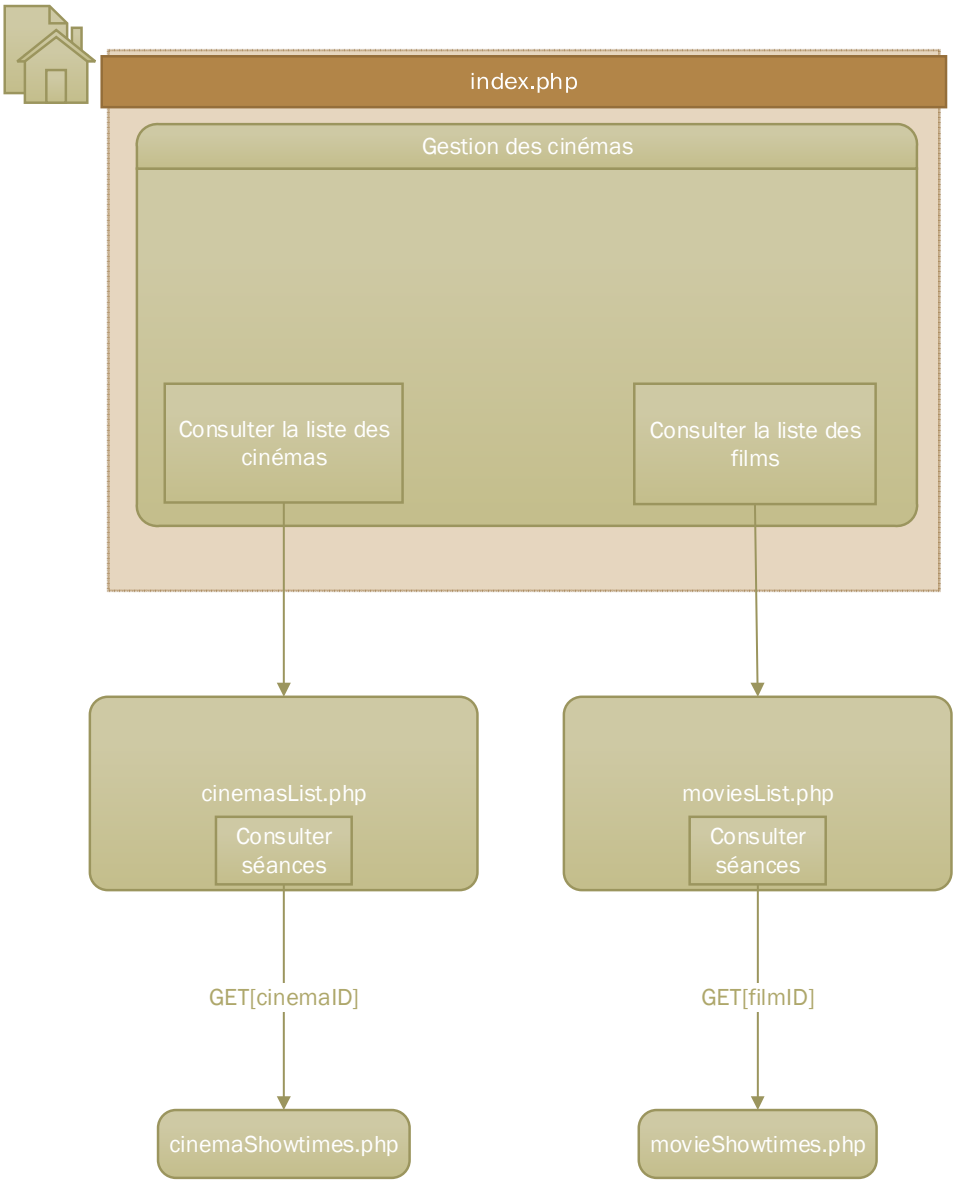
2. Gestion des cinémas

a) POC

(1) 1ère itération

N/A

(2) 2ème itération



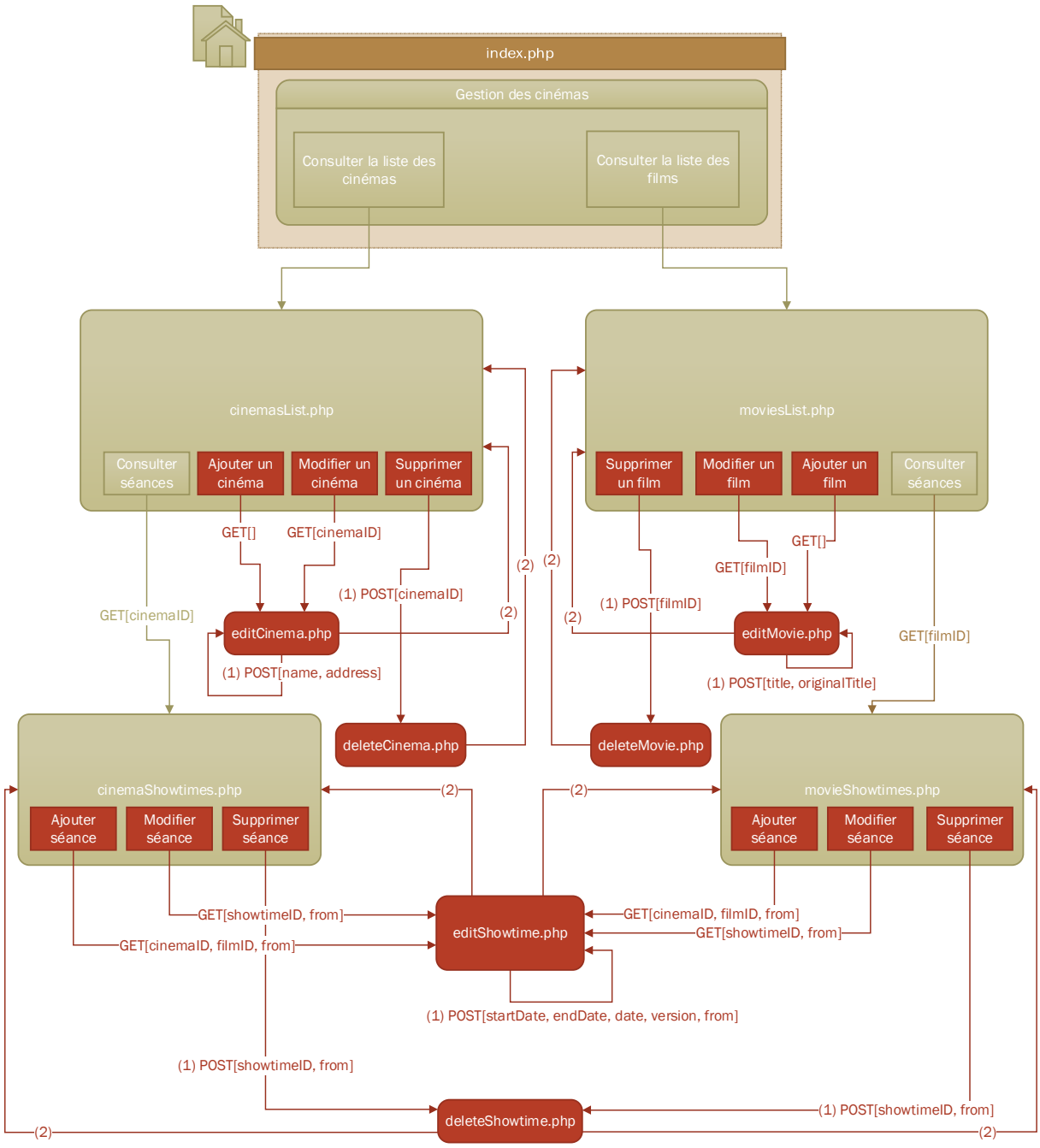
Légende :

——→ : redirection simple

——(1)——→ : (1) représente l'ordre des redirections

—POST []→ : POST[] représente la méthode de transfert des informations contenues dans []

(3) 3ème itération



Légende :

——► : redirection simple

——(1)——► : (1) représente l'ordre des redirections

—POST []► : POST[] représente la méthode de transfert des informations contenues dans []